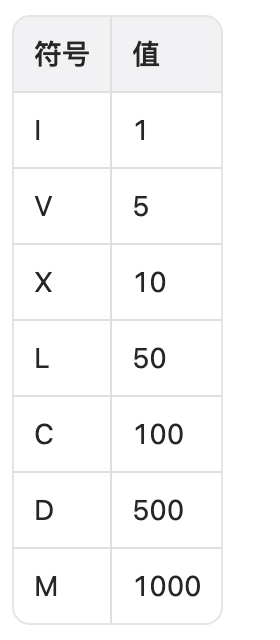
# 题目

七个不同的符号代表罗马数字，其值如下：



罗马数字是通过添加从最高到最低的小数位值的转换而形成的。将小数位值转换为罗马数字有以下规则：

如果该值不是以4或9开头，请选择可以从输入中减去的最大值的符号，将该符号附加到结果，减去其值，然后将其余部分转换为罗马数字。

如果该值以4或9开头，使用 减法形式，表示从以下符号中减去一个符号，例如4是5 (V)减1 (I): IV，9是10 (X)减1 (I)：IX。仅使用以下减法形式：4 (IV)，9 (IX)，40 (XL)，90 (XC)，400 (CD)和900 (CM)。

只有10的次方（I, X, C, M）最多可以连续附加3次以代表10的倍数。你不能多次附加5 (V)，50 (L)或500 (D)。如果需要将符号附加4次，请使用 减法形式。

给定一个整数，将其转换为罗马数字。

示例 1：

输入：num = 3749

输出： "MMMDCCXLIX"

解释：

3000 = MMM 由于 1000 (M) + 1000 (M) + 1000 (M)

700 = DCC 由于 500 (D) + 100 (C) + 100 (C)

40 = XL 由于 50 (L) 减 10 (X)

9 = IX 由于 10 (X) 减 1 (I)

注意：49 不是 50 (L) 减 1 (I) 因为转换是基于小数位

示例 2：

输入：num = 58

输出："LVIII"

解释：

50 = L

8 = VIII

示例 3：

输入：num = 1994

输出："MCMXCIV"

解释：

1000 = M

900 = CM

90 = XC

4 = IV

提示：

1 <= num <= 3999

# 分析

要将整数转换为罗马数字，核心思路是利用罗马数字的固定表示规则，通过预先定义所有可能的数值-符号对应关系（包括特殊的减法形式），从大到小逐步匹配并减去对应的值，拼接出最终的罗马数字字符串。

解题思路

1、罗马数字的表示规则：

罗马数字由特定符号组成，其中包含6种特殊减法形式（4、9、40、90、400、900）和其他常规加法形式。这些表示具有固定的数值，且按从大到小的顺序排列。

2、预定义数值与符号对应表：

将所有可能的数值（包括特殊减法形式）按从大到小的顺序与对应的罗马数字符号存储在数组中。例如：

{1000, "M"}, {900, "CM"}, {500, "D"}, {400, "CD"}, ..., {1, "I"}。

3、逐步匹配与拼接：

遍历预定义的数值-符号对，对于每个数值，若当前整数num大于或等于该数值，则将对应的符号拼接到结果中，并从num中减去该数值，重复此过程直到num变为0。

代码：

class Solution {

public:

string intToRoman(int num) {

// 预定义所有可能的数值-罗马数字对（从大到小排序）

vector<pair<int, string>> valueSymbols = {

{1000, "M"},

{900, "CM"},

{500, "D"},

{400, "CD"},

{100, "C"},

{90, "XC"},

{50, "L"},

{40, "XL"},

{10, "X"},

{9, "IX"},

{5, "V"},

{4, "IV"},

{1, "I"}

};

string result; // 存储最终的罗马数字

// 遍历所有数值-符号对，从大到小匹配

for (auto& pair : valueSymbols) {

int value = pair.first;

string symbol = pair.second;

// 若当前数值小于等于num，拼接符号并减去对应数值

while (num >= value) {

result += symbol;

num -= value;

}

// 若num已减为0，提前退出

if (num == 0) {

break;

}

}

return result;

}

};

代码解释

1、预定义数值-符号对：

数组valueSymbols按从大到小的顺序存储了所有可能的数值及其对应的罗马数字，包括特殊减法形式（如900对应"CM"、40对应"XL"等）。这种排序确保了我们总是先处理最大的可能数值，符合罗马数字“从高到低”的表示规则。

2、拼接罗马数字：

遍历valueSymbols中的每个数值-符号对，对于每个数值，只要num大于或等于该数值，就将对应的符号拼接到result中，并从num中减去该数值。例如，对于num=3749：

- 先匹配1000，拼接"MMM"，num变为749；

- 接着匹配500，拼接"D"，num变为249；

- 再匹配100，拼接"CC"，num变为49；

- 继续匹配40（"XL"）和9（"IX"），最终得到"MMMDCCXLIX"。

3、终止条件：

当num减为0时，说明所有数值已被转换，可提前退出循环，提高效率。

复杂度分析

- 时间复杂度：O(1)。由于num的范围是1到3999，预定义的数值-符号对数量固定（13对），且每个数值最多被处理几次（如1000最多处理3次，1最多处理3次），因此总操作次数是常数级。

- 空间复杂度：O(1)。除了存储结果的字符串外，仅使用固定大小的数组存储数值-符号对，空间开销不随num的变化而变化。